Assignemtn No. 3
DC

section a

```
gedit word_char.txt
Python is one of the most popular programming languages today, known for its simplicity,
extensive features and library support. Its clean and straightforward syntax makes it beginner-
friendly, while its powerful libraries and frameworks makes it perfect for developers. Python is:

A versatile, high-level programming language.
Easy-to-learn syntax, perfect for beginners and experts.
Known for its readability and extensive library support.
```

```
gedit reducer_char.py
#!/usr/bin/env python
"""reducer.py"""

import sys
current_char = None
current_count = 0
char = None

# input comes from STDIN
for line in sys.stdin:
# remove leading and trailing whitespace
    line = line.strip()
     # parse the input we got from mapper.py
    char, count = line.split('\t', 1)
    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
# count was not a number, so silently ignore/discard this line
        continue
    # this IF-switch only works because the output of the mapper is sorted by character
    if current_char == char:
        current_count += count
    else:
        if current_char is not None:
 # write result to STDOUT
```

```
            print('%s\t%s' % (current_char, current_count))
        current_count = count
        current_char = char

# Output the last character if needed
if current_char is not None:
    print('%s\t%s' % (current_char, current_count))
gedit mapper_char.py
#!/usr/bin/python
import sys
for line in sys.stdin:
    line = line.strip()
    # Iterate over each character in the line
    for char in line:
        if char.isalpha():  # Only count alphanumeric characters (you can remove this check if you
want all characters)
            print('%s\t%s' % (char, 1))
```

section b
gedit 1234_word.txt

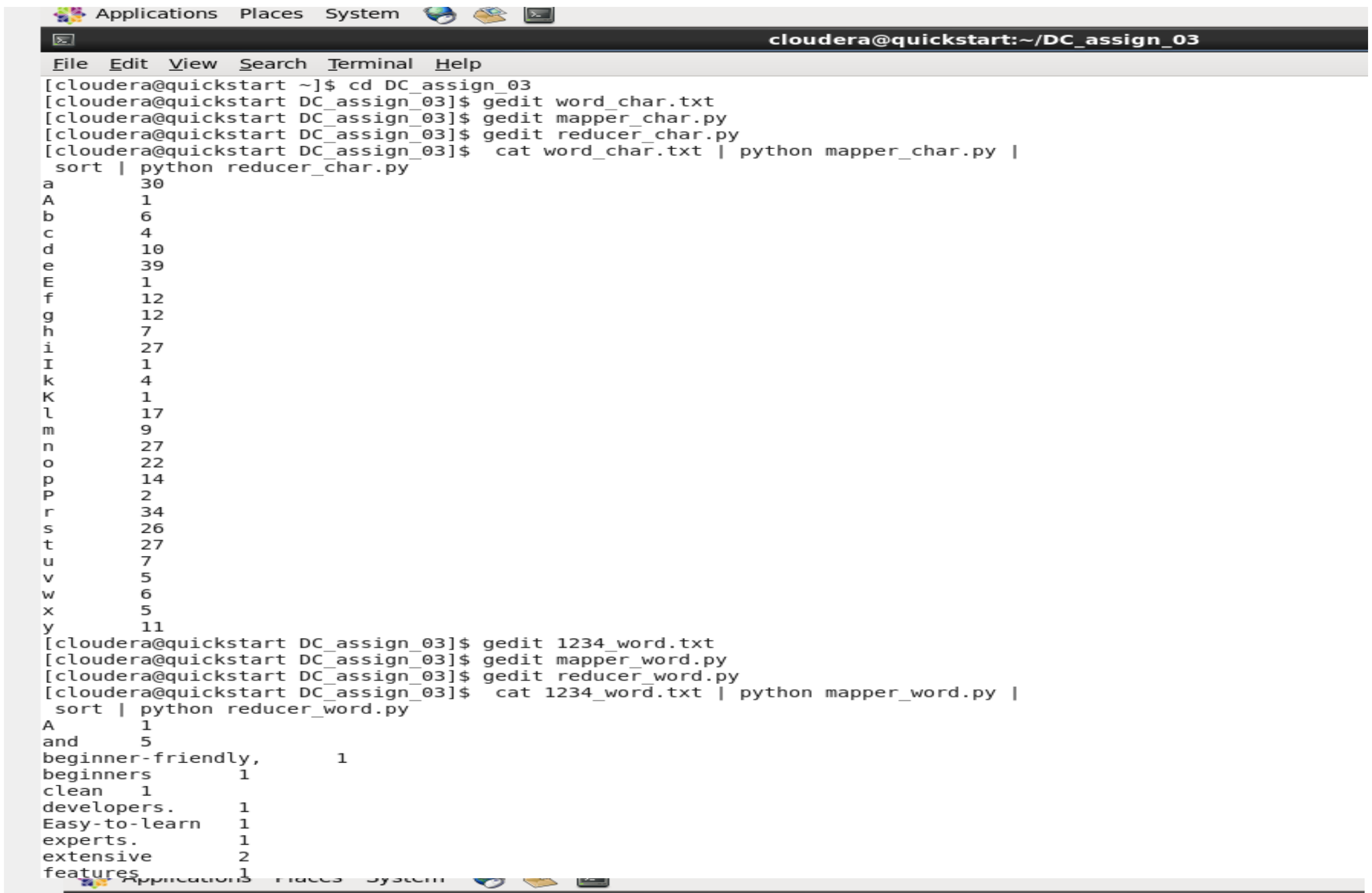
Python is one of the most popular programming languages today, known for its simplicity, extensive features and library support. Its clean and straightforward syntax makes it beginner-friendly, while its powerful libraries and frameworks makes it perfect for developers. Python is:

A versatile, high-level programming language.
Easy-to-learn syntax, perfect for beginners and experts.
Known for its readability and extensive library support.

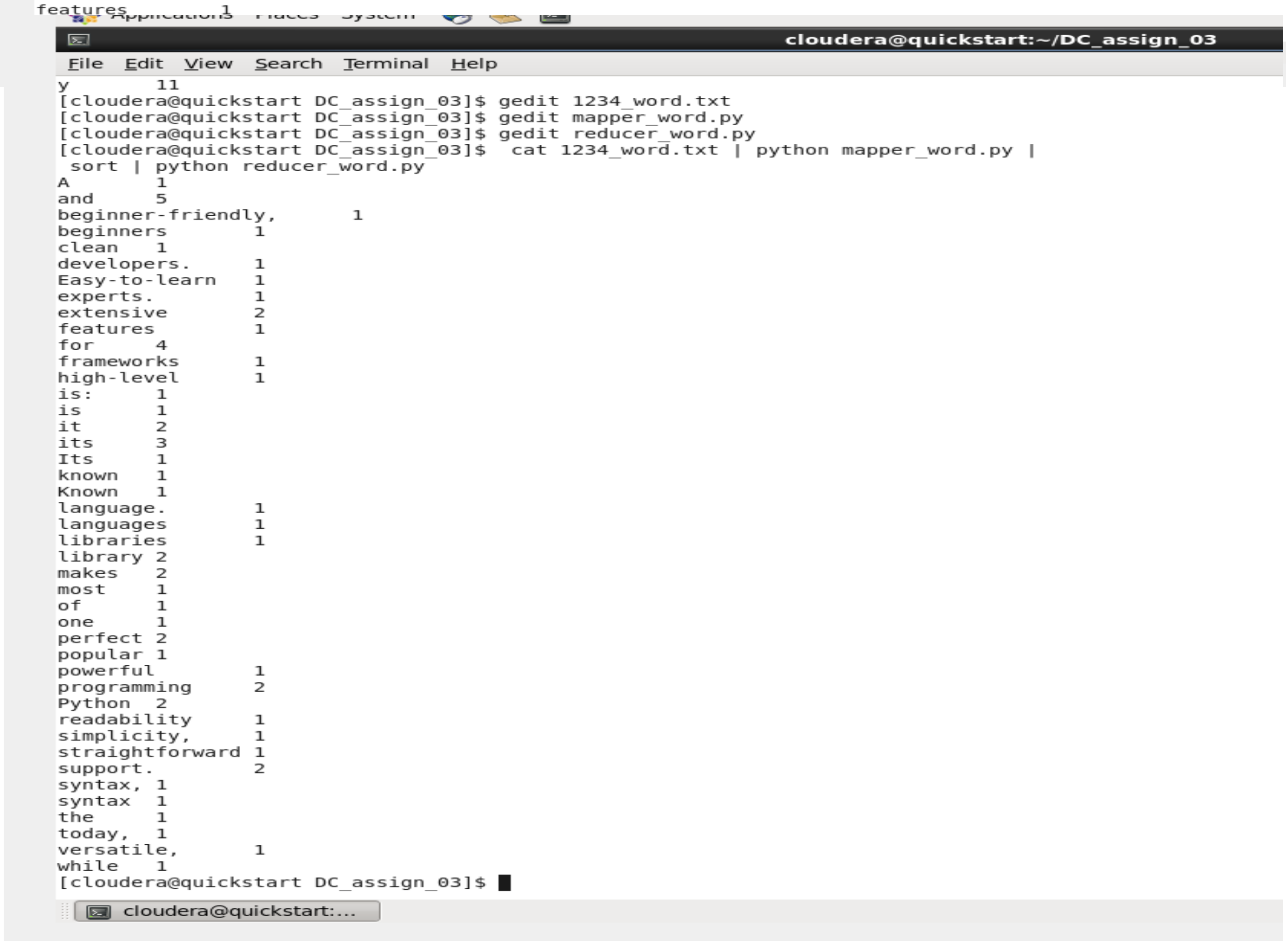
```
gedit reducer_word.py
#!/usr/bin/python
"""mapper.py"""

import sys
for line in sys.stdin:
line = line.strip()
words = line.split()
```

```
for word in words:
print '%s\t%s' % (word, 1)

gedit mapper_word.py
#!/usr/bin/env python
"""reducer.py"""

import sys
current_char = None
current_count = 0
char = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
# parse the input we got from mapper.py
    char, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently ignore/discard this line
        continue
  # this IF-switch only works because the output of the mapper is sorted by character
    if current_char == char:
        current_count += count
    else:
        if current_char:
            # write result to STDOUT
            print('%s\t%s' % (current_char, current_count))
        current_count = count
        current_char = char
# Output the last character if needed
if current_char == char:
    print('%s\t%s' % (current_char, current_count))
```

Output:


Applications Places System
cloudera@quickstart:~/DC_assign_03
File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ cd DC_assign_03
[cloudera@quickstart DC_assign_03]$ gedit word_char.txt
[cloudera@quickstart DC_assign_03]$ gedit mapper_char.py
[cloudera@quickstart DC_assign_03]$ gedit reducer_char.py
[cloudera@quickstart DC_assign_03]$  cat word_char.txt | python mapper_char.py |
 sort | python reducer_char.py
a       30
A       1
b       6
c       4
d       10
e       39
E       1
f       12
g       12
h       7
i       27
I       1
k       4
K       1
l       17
m       9
n       27
o       22
p       14
P       2
r       34
s       26
t       27
u       7
v       5
w       6
x       5
y       11
[cloudera@quickstart DC_assign_03]$ gedit 1234_word.txt
[cloudera@quickstart DC_assign_03]$ gedit mapper_word.py
[cloudera@quickstart DC_assign_03]$ gedit reducer_word.py
[cloudera@quickstart DC_assign_03]$  cat 1234_word.txt | python mapper_word.py |
 sort | python reducer_word.py
A       1
and     5
beginner-friendly,      1
beginners       1
clean   1
developers.     1
Easy-to-learn   1
experts.        1
extensive       2
features        1
```


cloudera@quickstart:~/DC_assign_03
File Edit View Search Terminal Help
cloudera@quickstart:...

```
y       11
[cloudera@quickstart DC_assign_03]$ gedit 1234_word.txt
[cloudera@quickstart DC_assign_03]$ gedit mapper_word.py
[cloudera@quickstart DC_assign_03]$ gedit reducer_word.py
[cloudera@quickstart DC_assign_03]$  cat 1234_word.txt | python mapper_word.py |
 sort | python reducer_word.py
A       1
and     5
beginner-friendly,      1
beginners       1
clean   1
developers.     1
Easy-to-learn   1
experts.        1
extensive       2
features        1
```

```
for     4
frameworks      1
high-level      1
is:     1
is      1
it      2
its     3
Its     1
known   1
Known   1
language.       1
languages       1
libraries       1
library 2
makes   2
most    1
of      1
one     1
perfect 2
popular 1
powerful        1
programming     2
Python  2
readability     1
simplicity,     1
straightforward 1
support.        2
syntax, 1
syntax  1
the     1
today,  1
versatile,      1
while   1
[cloudera@quickstart DC_assign_03]$ 
```